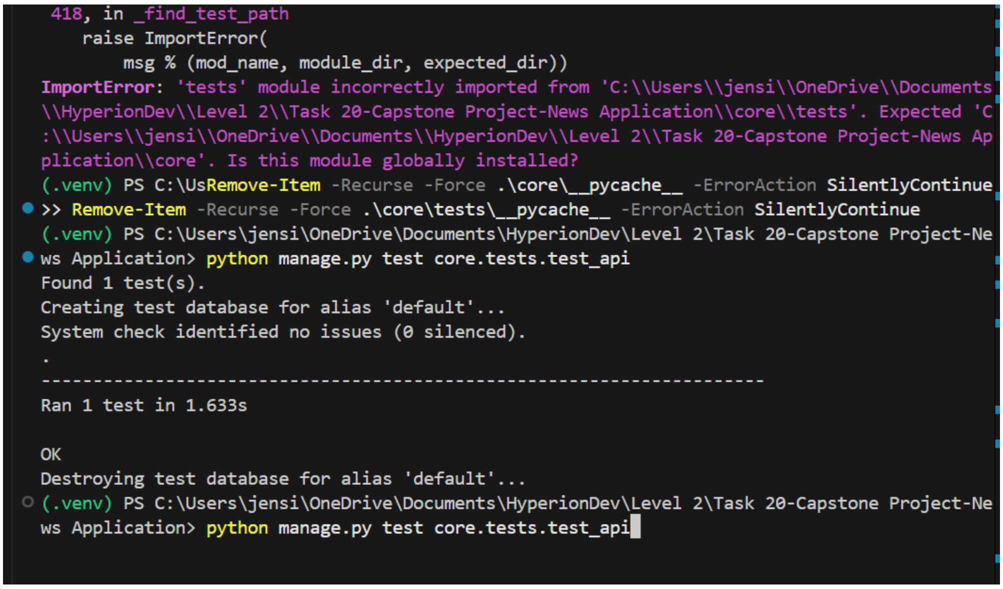
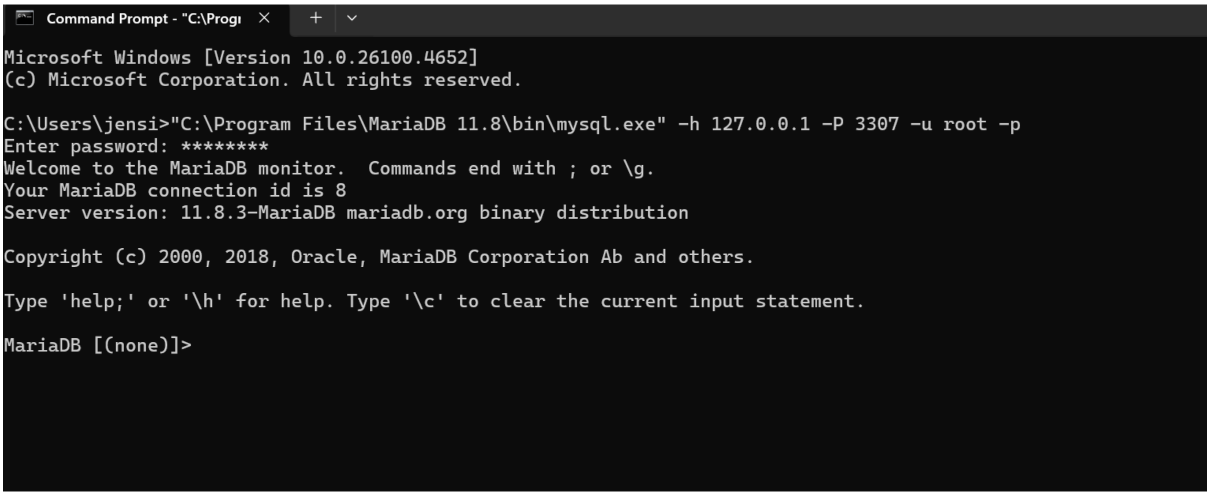
Report for Task 20-

Screenshots:



```
 418, in _find_test_path
    raise ImportError(
        msg % (mod_name, module_dir, expected_dir))
ImportError: 'tests' module incorrectly imported from 'C:\\Users\\jensi\\OneDrive\\Documents
\\HyperionDev\\Level 2\\Task 20-Capstone Project-News Application\\core\\tests'. Expected 'C
:\\Users\\jensi\\OneDrive\\Documents\\HyperionDev\\Level 2\\Task 20-Capstone Project-News Ap
plication\\core'. Is this module globally installed?
(.venv) PS C:\UsRemove-Item -Recurse -Force .\core\__pycache__ -ErrorAction SilentlyContinue
>> Remove-Item -Recurse -Force .\core\tests\__pycache__ -ErrorAction SilentlyContinue
(.venv) PS C:\Users\jensi\OneDrive\Documents\HyperionDev\Level 2\Task 20-Capstone Project-Ne
ws Application> python manage.py test core.tests.test_api
Found 1 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
.
----------------------------------------------------------------------
Ran 1 test in 1.633s

OK
Destroying test database for alias 'default'...
(.venv) PS C:\Users\jensi\OneDrive\Documents\HyperionDev\Level 2\Task 20-Capstone Project-Ne
ws Application> python manage.py test core.tests.test_api
```



```
Microsoft Windows [Version 10.0.26100.4652]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jensi>"C:\Program Files\MariaDB 11.8\bin\mysql.exe" -h 127.0.0.1 -P 3307 -u root -p
Enter password: ********
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 8
Server version: 11.8.3-MariaDB mariadb.org binary distribution

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

The News Portal Web Application is a Django-based platform designed to allow publishers, editors, journalists, and readers to interact in a structured news environment. It enables journalists to create and manage articles, editors to review and approve content, and readers to access published news based on their subscriptions. The system implements a custom user model with role-based permissions to ensure secure and controlled access. In addition to a web interface, the application offers a RESTful API for third-party integration and uses MariaDB for reliable, scalable data storage.

The objective of this project is to develop a functional and user-friendly news management system that supports different roles within a newsroom. It aims to streamline the process of creating, editing, approving, and publishing articles while ensuring appropriate access control for each user type. By integrating a MariaDB database and Django's built-in authentication framework, the project seeks to deliver a secure and efficient workflow. Additionally, the inclusion of an API provides flexibility for future expansion, such as mobile app integration or third-party news aggregators.

The development of the News Portal Web Application followed a structured approach, starting with the setup of the Django framework and a custom project configuration. A custom user model was implemented to support role-based permissions for editors, journalists, and readers. The application's architecture was divided into logical components, including models for managing publishers, articles, and user roles, as well as views and templates for rendering content. Django's admin interface was customized to allow administrators to manage users and content efficiently. MariaDB was configured as the database backend to ensure high performance and reliability. The REST API was developed using Django REST Framework (DRF) to enable external data access. Throughout development, unit testing was performed to validate core functionality, and sample data was created to verify end-to-end workflows.

The completed News Portal Web Application successfully meets the project requirements by providing a fully functional, role-based news management system. Editors can review and approve articles submitted by journalists, while readers can browse published content through an intuitive interface. The system enforces role-specific permissions, ensuring that each user only has access to the actions relevant to their role. Integration with MariaDB allows for efficient data storage and retrieval, while the REST API enables seamless external access to news data. The admin panel provides robust content management capabilities, and all major workflows, from user creation to article publishing, were tested and confirmed to operate as intended.

This project successfully delivered a secure, scalable, and role-based News Portal Web Application built with Django and powered by a MariaDB backend. By implementing distinct permissions for editors, journalists, and readers, the system ensures a structured workflow and maintains content quality. The integration of a REST API future-proofs the application by allowing expansion into other platforms and services. Throughout development, emphasis was placed on maintainable code, robust testing, and clear documentation. The result is a reliable platform that can be adapted for real-world newsroom operations, demonstrating both the practicality and flexibility of Django for content management solutions.